



Appendix  
Appl. No. 09/787,648  
Amdt. dated Oct. 19, 2005  
Reply to Office Action of June 24, 2005  
Marked-Up Specification

**METHOD AND DEVICE FOR CRYPTOGRAPHICALLY PROCESSING DATA**

[0001] BACKGROUND OF THE INVENTION

[0002] 1. Field of the Invention

5 [0003] The invention relates to a method for  
cryptographically processing data, comprising feeding, to a  
cryptographic process, values, namely, the data and a key,  
and carrying out the process in order to form  
cryptographically processed data. ~~Such method is generally~~  
10 ~~known.~~

[0004] 2. Description of the Prior Art

[0005] For cryptographically processing data, in  
15 practice there are often applied generally known processes.  
Examples of such cryptographic processes (algorithms) are  
DES and RSA [DES = Data Encryption Standard and RSA =  
Rivest, Shamir & Adleman], which are described, e.g., in  
the book "Applied Cryptography" by B. Schneier (2nd  
20 edition), New York, 1996.

[0006] ~~Said~~ These processes are published since it was  
assumed that, in the event of sufficiently large key  
lengths, it would be impossible, on the basis of the  
25 processed data, to retrieve the original data and/or the  
key, even if the cryptographic process were known.

[0007] However, Encryptographic algorithms can be attacked -the goal always is to find the encryption key in use- in different ways: (1) Mathematical attacks like differential and linear cryptanalysis; (2) Hardware oriented attacks, called "Side Channel Attacks", viz. attacks based on power consumption analysis or I/O timing analysis.

[0008] U.S. Patent No. 5,745,577 discloses a method for advanced key scheduling of a secret key. The aim is to offer a protection against ~~said~~ mathematical attacks (differential and linear cryptanalysis) by amending the encryption algorithm. Amending the algorithm will cause change of its output and thus the disclosed method does not present any improvement against ~~said~~ the "Side Channel Attacks".

[0009] SUMMARY OF THE INVENTION

[0010] The present invention aims to improve the protection of a cryptographic device against "Side Channel Attacks". In short, ~~said~~ the improvement is achieved by masking the data and/or the key by means of generating extra, auxiliary input (data or key) and compensating its influence to the output by adding, to the "main" encryption process, an auxiliary (compensating) process. By ~~said~~ these masking measures, it will be much more difficult to derive the value of data or key from the behavior of the power consumption of the cryptographic device. ~~Said~~ this masking, however, happens in such a way that the result of the process as a whole remains

unchanged: with the same input and key the amended algorithm results into the same, unchanged output.

5 [0011] Thus, the invention presents an inventive  
~~method of the type referred to in the preamble according~~  
~~to the invention~~ which is characterized by feeding, to  
the process, auxiliary values, while compensating, by  
means of an auxiliary process, the influence of the  
auxiliary values to the output data, in order to mask  
10 the values used in the process.

[0012] By masking the data and/or key(s), it becomes  
considerably more difficult to derive ~~said~~ their values  
on the basis of the behavior of the process. The result  
15 of the process, i.e., the collection of processed data,  
in the event of a suitable choice of the auxiliary  
values may be unchanged, i.e., identical to the result  
of the process, if no auxiliary values have been fed to  
it. In this connection, an "auxiliary value" is  
20 understood to mean a value (data or key) which is fed to  
the process as a supplement to the corresponding data  
and key.

[0013] The invention is therefore based on the  
25 insight that the derivation of the values used in a  
cryptographic process is rendered considerably more  
difficult if ~~said~~ those values are masked using ~~said~~ the  
auxiliary values and ~~said~~ the auxiliary process.

30 [0014] The invention is partly based on the further  
insight that the use of auxiliary values does not  
necessarily affect the outcome of the process.

[0015] In a first embodiment of the invention, an auxiliary value comprises a supplementary key which is fed to a supplementary process in order to form the key.

5

[0016] By applying a combination of a known process and a supplementary process, there is formed a new cryptographic process, unknown per se, even if the supplementary process is also known per se.

10

[0017] By deriving the key used for the known process (primary key) from a supplementary key (secondary key) using a supplementary process, there is achieved that not the (primary) key of the known process but the supplementary (secondary) key is offered to the combination of processes. In other words, externally the supplementary (secondary) key, and not the real (primary) key of the process proper, is used. Derivation of the key from the original data and the processed data has thereby become impossible. In addition, the derivation of the supplementary key has been rendered seriously more difficult, since the combination of the original process and the supplementary process is not known.

15

20

25

[0018] ~~Said~~The embodiment of the invention is therefore based, *inter alia*, on the insight that prior knowledge of a cryptographic process is undesirable, such is contrary to what was so far assumed. ~~Said~~The embodiment is also based on the ~~F~~further insight that attacks which elaborate on knowledge of the process become considerably more difficult if the process is unknown.

30

[0019] The supplementary process preferably comprises a cryptographic process. This renders the derivation of the supplementary key more difficult. Basically, however, a simple encoding may be applied, e.g., as a supplementary process. In the event of a cryptographic process, there is preferably applied an auxiliary key.

[0020] The supplementary process advantageously is an invertible process. This enables the application of the method according to the invention in existing equipment with minimum modifications. If, e.g., a first device gives off a (supplementary) key which is applied in a second device according to the invention, then in the first device there may be used the inverse of the supplementary process to derive the supplementary key from the original key. In other words, although in both the first and the second devices internally the original (primary) key is used, there is exchanged, between the devices, the supplementary (secondary) key. Intercepting the supplementary key, however, does not result in knowledge of the original key.

[0021] It may be advantageous, if carrying out the supplementary process takes place exclusively, ~~if that~~ the data has predetermined properties. In this manner, cryptographic processing may be carried out for specific, selected data only, while such is blocked for all other data. In this manner, there is achieved a supplementary protection.

[0022] ~~An optimum~~ Further enhanced security is provided if the process and the supplementary process are

each constructed of several steps and in which there are alternately carried out steps of the process and the supplementary process. As a result, the properties of the known process are further veiled, as a result of which the derivation of the keys is further complicated.

[0023] In a second embodiment of the invention, the process comprises several steps, each of which has a cryptographic operation for processing right-hand data derived from the data and a combinatory operation for combining, with the left-hand data derived from the data, the processed right-hand data in order to form modified left-hand data, in which the right-hand data, prior to the first step, is combined with a primary auxiliary value and the left-hand data is combined with an additional auxiliary value. As a result, the data used in the steps and transferred between the steps is masked.

[0024] In order ~~to make it possible~~ for the primary and additional auxiliary values ~~do not make themselves felt in~~ not to overly affect the end result of the process, the right-hand data is combined, preferably immediately after the last step, with a further primary auxiliary value, and the modified left-hand data is combined with a further additional auxiliary value.

[0025] In order ~~not to have that~~ the result of the operations is not affected by the primary auxiliary values, the method according to the invention is preferably carried out in such a manner that the right-hand data, in each step and prior to the operation, is combined with the primary auxiliary value of ~~said that~~

step. A further protection is achieved if the processed right-hand data, following the processing, is combined with a secondary auxiliary value of ~~said~~that step.

5 [0026] The secondary auxiliary value of a given step is advantageously formed from the combination of the primary auxiliary value of the preceding step and the primary auxiliary value of the next step.

10 [0027] As a result, it becomes possible to compensate the auxiliary value in the repeatedly next step, as a result of which ~~said~~the auxiliary value will not ~~make itself felt in overly affect~~ the end result of the process.

15 [0028] It is possible to carry out the method according to the invention in such a manner, that all primary auxiliary values are equal. As a result, a very simple practical realization is possible. The use of  
20 several auxiliary values, which are preferably random numbers and are newly generated ~~anew for each~~ time the process is carried out, however, offers a greater cryptographic security.

25 [0029] A further simplification of ~~said~~the embodiment may be obtained if the primary auxiliary values and/or secondary auxiliary values repeatedly have been combined in advance with the operation in question. This is to say, combining with auxiliary values is processed in the  
30 operation in question (e.-g., a substitution), in such a manner that the result of ~~the~~that operation ~~in question~~ is equal to that of the original operation plus one or

two combinatory operations with auxiliary values. By in advance including in the operation the combinatory operations, a ~~more simple~~ simpler and faster practical realization is possible.

5

[0030] ~~Said~~ Such combinatory operations are preferably carried out using an XOR operation [XOR = eXclusive OR]. Other combinatory operations, however, such as binary adding, are basically possible as well.

10

[0031] The invention further provides a circuit for carrying out a method for cryptographically processing data. In addition, the invention supplies a payment card and a payment terminal provided with such circuit.

15

[0032] Below, the invention will be further explained on the basis of the exemplary embodiments shown in the figures.

20

[0033] BRIEF DESCRIPTION OF THE DRAWINGS

[0034] FIG. 1 schematically shows a cryptographic process according to the prior art.

25

[0035] FIG. 2 schematically shows a first cryptographic process according to a first embodiment of the invention.

30

[0036] FIG. 3 schematically shows a second cryptographic process according to a first embodiment of the invention.



[0037] FIG. 4 schematically shows a way in which- the processes of figures FIG. 1 and 2 may be carried out.

[0038] FIG. 5 schematically shows a cryptographic process having several steps according to the prior art.

[0039] FIG. 6 schematically shows a first cryptographic process according to a second embodiment of the invention.

[0040] FIG. 7 schematically shows a second cryptographic process according to a second embodiment of the invention.

[0041] FIG. 8 schematically shows a third cryptographic process according to a second embodiment of the invention.

[0042] FIG. 9 schematically shows a circuit in which the invention is applied.

[0043] FIG. 10 schematically shows a payment system in which the invention is applied.

[0044] PREFERRED EMBODIMENTS

[0045] A (cryptographic) process P according to the prior art is schematically shown in FIG. 1. To the process P, there are fed input data X and a key K. On the basis of the key K, the process P converts the input data X into (cryptographically) processed output data Y:  $Y = PK(X)$ . The process P may be a known cryptographic

process, such as DES (Data Encryption Standard), triple DES, or RSA (Rivest, Shamir & Adleman).

5 [0046] If the input data X and the output data Y are known, it is basically possible to derive the key K used. In the event of a key of sufficient length (i.e., a sufficient number of bits), it was so far deemed impossible to derive ~~said~~the key, even if the process P were known. Impossible in this case is to say that in 10 theory it is admittedly possible, e.-g., by trying out all possible keys, to retrieve the key used, but that such requires an impossibly long computational time. Such "brute-force attack" is therefore hardly a threat to the cryptographic security.

15 [0047] Attacks recently discovered, however, make use of knowledge of the process, as a result of which the number of possible keys may be reduced drastically. Deriving the key K used and/or the input data X from the 20 output data Y therefore becomes possible within acceptable computational times.

25 [0048] The principle of the invention, whose object it is to render such attacks considerably more difficult and time-consuming, is schematically shown in FIG. 2. Just as in FIG. 1, to a (known) process P there are fed input data X and a (secret) key K to generate output data Y.

30 [0049] Contrary to the situation of FIG. 1, in the situation of FIG. 2 the key K is fed to the process P from a supplementary process P\*. The supplementary process P\* has a supplementary (secondary) key K\* as

input data to produce, under the influence of an auxiliary key  $K'$ , the (primary) key  $K$  as output data. The key  $K$  is therefore not fed, as is the case in the situation of FIG. 1, from an external source (e.g., a memory) to the process  $P$ , but is produced by the process  $P^*$  from the supplementary (secondary) key  $K^*$ :

$$K = P^*_{K'}(K^*).$$

[0050] It is therefore the secondary key  $K^*$ , instead of the primary key  $K$ , which is predetermined and stored, e.g., in a key memory (not shown). According to the invention, the primary key  $K$ , which is fed to the process  $P$ , is not predetermined.

[0051] The auxiliary key  $K'$  may be a permanently stored, predetermined key. It is also possible to apply a supplementary process  $P^*$  in which no auxiliary key  $K'$  is used.

[0052] The combination of the processes  $P$  and  $P^*$  forms a new process which is schematically designated by  $Q$ . To the process  $Q$  which, on account of the supplementary process  $P^*$ , is unknown per se, the input data  $X$  and the (secondary) key  $K^*$  are fed to produce the output data  $Y$ . The relationship between the secondary key  $K^*$  and the primary key  $K$  is veiled by the supplementary process  $P^*$ .

[0053] The supplementary process  $P^*$  preferably is the inverse of another, invertible process  $R$ . This is to say:

$$P^* = R^{-1} .$$

[0054] This enables producing the secondary key  $K^*$  from the primary key  $K$  using  $R$  and the auxiliary key  $K'$ —:

5

$$K^* = R_{K'}(K) ,$$

as will be further explained later by reference to FIG.

5. The new process  $Q$  may possibly be extended by the  
 10 process  $R$ , in such a manner that the primary key  $K$ , instead of the secondary key  $K^*$ , is fed to the process  $Q$ . The primary key  $K$  in this case in the process  $Q$  is derived from:

15

$$K = P^*_{K'}(K^*) = P^*_{K'}(R_{K'}(K)) .$$

This enables using the same (primary) key as in the prior art.

20

[0055] The cryptographic process  $Q$  according to the invention, schematically shown in FIG. 3, also comprises a process  $P$  having a primary key  $K$  and a supplementary process  $P^*$  having an auxiliary key  $K'$ , the primary key  $K$  being derived from the supplementary key  $K^*$  by the  
 25 supplementary process  $P^*$ . Supplementing the process of FIG. 1, in this case the input data  $X$  is also fed to the supplementary process  $P^*$ , in such a manner that the primary key  $K$  is determined partly as a function of the input data  $X$ :

30

$$K = P^*_{K'}(K^*, X) .$$

[0056] As a result, there is obtained a supplementary cryptographic protection. In addition, as a result the possibility is offered to carry out the supplementary process P\* exclusively if certain input data is offered.

5 This is to say that the supplementary process P\* may comprise a test of the input data X, and carrying out the supplementary process P\* may depend on the result of ~~said~~ that test. Thus, the supplementary process P\*, e.g., may be carried out only if the last two bits of the input  
10 data X equal zero. The effect of such an input data-dependent operation is that only for certain input data X the correct primary key K will be produced in such a manner that only ~~said~~ that input data will deliver the desired output data Y. It will be understood that as a  
15 result the cryptographic security is further enhanced.

[0057] FIG. 4 schematically shows the way in which substeps of the processes P and P\* may be carried out alternately ("interleaving") in order to further  
20 enhance the protection against attacks. The substeps may include so-called "rounds", such as, e.g., in the case of DES. The substeps, however, preferably comprise only one or a few instructions of a program, with which the processes are being carried out.

25 [0058] In a first step 101, there is carried out a first substep P<sub>1</sub> of the process P. Subsequently, in a second step 102, the first substep P<sub>1</sub>\* of the supplementary process P\* is carried out.

30 [0059] Likewise, in a third step 103, the second substep P<sub>2</sub> of the process P is carried out etc. This

continues until, in step 110, the last substep  $P_n^*$  of the supplementary process  $P^*$  has been carried out, it being assumed, for the sake of the example, that the processes  $P$  and  $P^*$  comprise an equal number of substeps. If such  
5 is not the case, in step 110 there is carried out the last corresponding substep, and in further steps the remaining substeps are carried out.

[0060] By alternating the substeps of the process  $P$ , which is known per se, and the process  $P^*$  (possibly known per se as well), there may be obtained a series of substeps which does not correspond to that of a known process. As a result, the nature of the process is more difficult to recognize.

15 [0061] The cryptographic process  $P$  schematically shown, only by way of example, in FIG. 5, according to the prior art comprises several steps  $S_i$  (i.e.,  $S_1, S_2, \dots, S_n$ ). In each step  $S_i$ , (right-hand) data  $RD_i$  is  
20 fed to a cryptographic operation  $F_i$ . ~~Said~~ The cryptographic operation may itself comprise a number of substeps, such as an expansion, a combination with a key, a substitution and a permutation which, however, have not been designated separately for the sake of the simplicity  
25 of the drawing. The cryptographic operation  $F_i$  provides processed data  $FD_i$ :

$$FD_i = F_i(RD_i).$$

30 In a combinatory operation  $CC_i$  ( $CC_1, CC_2, \dots$ , the index  $i$  always indicating the step  $S$  in question), the processed data  $FD_i$  —is combined with left-hand data  $LD_i$  to form

modified (left-hand) data  $SD_i$  which, just as the original right-hand data  $RD$ , is passed on to the next step. The combinatory operations  $CC_i$  preferably are XOR operations (symbol:  $\oplus$ ).

5

[0062] As is shown in FIG. 5, at the end of each step  $S_i$  the modified left-hand data  $SD_i$  and the right-hand data  $RD_i$  change positions in such a manner that they form the right-hand data  $RD_{i+1}$  and the left-hand data  $LD_{i+1}$  of the next step  $S_{i+1}$ .

10

[0063] The left-hand data  $LD_1$  and the right-hand data  $RD_1$  of the first step  $S_1$  were derived, in a preceding operation, from input data  $X$  and, in doing so, may undergo a preparatory processing, such as an input permutation. The output data  $SD_n$  and  $RD_n$  of the last step  $S_n$  form the processed data  $Y$  of the process  $P$ , possibly after it has undergone a final operation, such as an output permutation  $PP^{-1}$ .

15

20

[0064] The cryptographic process of FIG. 6 largely corresponds to that of FIG. 5. In accordance with the invention, the data present in and between the steps is masked with auxiliary values. For this purpose, in this embodiment the first step  $S_1$  is preceded by (preparatory) combinatory operations  $DC$  and  $EC$ , which are preferably XOR operations as well. They combine the left-hand data  $LD_1$ , and the right-hand data  $RD_1$ , respectively, which originate from the preparatory processing operation ~~operation~~  $(PP)$ , with a zeroth auxiliary value  $A_0$  and a first auxiliary value  $A_1$ . The results of the combinatory operations  $DC$

25

30

and EC are left-hand masked data  $LD'_1$  and right-hand masked data  $RD'_1$ , respectively (in the continuation of this text, masked data will be designated by an apostrophe). The maskings make themselves felt in the subsequent steps. Since the left-hand data of the second step  $S_2$  is equal to the masked right-hand data of the first step  $S_1$ , ~~said~~ then the left-hand data  $LD'_2$  is masked as well. The right-hand data  $-RD'_2$  of the second step is masked since it is equal to the masked, modified data  $SD'_1$ .

[0065] Combining the data  $LD_i$  and  $RD_i$  with the auxiliary values  $A_i$  therefore results in the modified data  $LD'_i$  and  $RD'_i$  being masked, as a result of which it is considerably more difficult to derive the original data  $X$  or the key used from the masked data  $LD'_i$  and  $RD'_i$ .

[0066] In order to remove the auxiliary values  $A_i$  prior to the final operation ( $PP^{-1}$ ), there are provided completing combinatory operations FC and GC, which combine the modified and masked left-hand data  $SD'_n$  of the last step  $S_n$  with an auxiliary value  $A_{n+1}$  and the masked right-hand data  $RD'_n$  with an auxiliary value  $A_n$ , respectively. On account of  $A_i \oplus A_i$  being zero in this manner the maskings are removed by the auxiliary values  $A_i$ . As a result, it is possible to carry out the method in such a manner that, notwithstanding the use of the auxiliary values  $A_i$ , the final data  $Y$  is equal to that which would have been obtained by the conventional method according to FIG. 5.



[0067] In order to exclude the effect of the auxiliary values  $A_i$  on the results  $FD_i$  of the operations  $F_i$ , in each step  $S_i$  there is preferably present a supplementary combinatory operation  $AC_i$  which combines the right-hand data  $RD_i$  with a (primary) auxiliary value  $A_i$  before this data is fed to the cryptographic operation  $F_i$ . The result of each supplementary combinatory operation  $AC_i$  is non-masked right-hand data  $RD_i$ , so that the cryptographic operation  $F_i$  works on the same data as in the process of FIG. 5.

[0068] There may be advantageously inserted a further combinatory operation  $BC_i$  between the cryptographic operation  $F_i$  and the combinatory operation  $CC_i$  with the purpose of combining the processed (right-hand) data  $FD_i$  with a further (secondary) auxiliary value  $B_i$ . As a result, there may be achieved a masking of the processed data  $FD_i$  and a further masking of the (modified) left-hand data  $SD'_i$ . The combinatory operations  $AC_i$  and  $BC_i$  preferably are XOR operations as well.

[0069] In accordance with a further aspect of the invention, the auxiliary values  $A_i$  and  $B_i$  are related. The secondary auxiliary values  $B_i$  are formed, preferably using an XOR operation, from the first auxiliary value  $A_{i-1}$  of the previous step and the auxiliary value  $A_{i+1}$  of the next step:

$$B_i = A_{i-1} \oplus A_{i+1}$$

This results in each primary auxiliary value  $A_{i+1}$  which, using a further supplementary combinatory operation  $BC_i$ , is combined with the processed right-hand data  $FD_i$  as an

ingredient of the secondary auxiliary value  $B_i$ , repeatedly being compensated in the next step, i.e., step  $S_{i+1}$ , by means of a combinatory operation  $AC_i$  before the right-hand data  $RD_{i+1}$  is subjected to the operation  $F_i$ . The (masked) right-hand data  $RD'_i$  in question, which forms the (masked) left-hand data  $LD'_{i+1}$  of the still next step  $S_{i+2}$  are combined there with the primary auxiliary value  $A_{i+1}$  and is compensated in this manner. The auxiliary value  $A_{i+1}$  makes itself felt in the modified data  $SD'_i$ , in such a manner that this remains masked between two steps.

[0070] The left-hand data  $LD'_1$  of the first step  $S_1$  is masked with the additional or zeroth (primary) auxiliary value  $A_0$ . By combining, with the secondary auxiliary value  $B_1 = A_0 \oplus A_2$ , the initial auxiliary value  $A_0$  is removed (on account of  $A_0 \oplus A_0$  being zero), but the auxiliary value  $A_2$  and the masking achieved therewith are maintained. The zeroth auxiliary value  $A_0$  in this embodiment is preferably chosen equal to the first auxiliary value  $A_1$ .

[0071] Although all primary auxiliary values  $A_i$  are preferably chosen to be different, with the exception of  $A_0 = A_1$ , it is possible to choose all primary auxiliary values  $A_i$  to be equal. In this case, all secondary auxiliary values  $B_i$  in the embodiment shown will be equal to zero, so that the further combinatory operations  $BC_i$  may be omitted. The invention further applies to processes  $P$  which contain only one step  $S$ , or have a deviating structure.

[0072] In the process of FIG. 7, which largely corresponds to that of FIG. 6, the combinatory operations

AC<sub>i</sub> and BC<sub>i</sub> and the cryptographic operation F<sub>i</sub> in each step are integrated to form a combined operation F'<sub>i</sub>. Integrating the combinatory operations in the operations F<sub>i</sub> is possible by suitably adjusting, e.g., a substitution table of the operation F<sub>i</sub>. As a result, the supplementary combinatory operations AC<sub>i</sub> and BC<sub>i</sub> may be omitted and the result of the adjusted operation F'<sub>i</sub> is equal to the result of the total of the operation F<sub>i</sub> proper and the combinatory operations:

$$FD'_i = F'_i (RD'_i) = B_i \oplus F_i (A_i \oplus |RD'_i|).$$

[0073] Basically, each step S<sub>i</sub> requires a different combinatory operation F<sub>i</sub> in which various auxiliary values A<sub>i</sub> are integrated (see FIG. 6). Only if the auxiliary values A<sub>i</sub> are chosen equal, i.e., A<sub>1</sub>=A<sub>2</sub>=...=A<sub>n</sub>, the combinatory operations F<sub>i</sub> in this embodiment may be equal.

[0074] Each time the process is carried out, the values A<sub>i</sub> are preferably chosen anew. For the process of FIG. 7, this means that the combined operations F'<sub>i</sub> are then determined anew. Since the operations F'<sub>i</sub> in many implementations will comprise the use of several tables, such as substitution tables, said tables will be determined anew each time the process P is carried out. In order to offer a supplementary protection against attacks, according to a further aspect of the invention the tables will be determined in random order. If a combined operation F'<sub>i</sub> comprises, e.g., eight tables, ~~said these~~ eight tables will be determined in another order each time ~~said the~~ operation F'<sub>i</sub> is carried out ~~a new. Said anew.~~ The order may be determined on the basis of the contents of an order

register, which contents may each time be formed by a random number originating from a random-number generator. On the basis of the contents of the order register there may each time be composed a fresh lookup table. Using the  
5 lookup table, the tables may be written to a memory and later be read out.

[0075] According to a further aspect of the invention, supplementing this or instead thereof, the elements of each  
10 table may be determined and/or stored in random order. With this measure, improved ~~it is achieved that the~~ protection against attacks is also improved ~~achieved~~. In this case, too, there may be applied a lookup table on the basis of which the elements may later be retrieved.

15 [0076] The measures referred to above may also be applied in another embodiment of the invention, such as the one of FIG. 8, or in completely different other processes, whether cryptographic or not.

20 [0077] The embodiment of FIG. 8 largely corresponds to that of FIG. 7. Supplementing FIG. 7, each step  $S_i$ , with the exception of the last step  $S_n$ , includes a combinatory operation  $HC_i$  which combines the right-hand data  $RD'_i$  with a  
25 tertiary auxiliary value  $W_i$ . The tertiary auxiliary value  $W_i$  preferably equals the XOR combination of the auxiliary values  $A_0$  and  $A_i$ :

$$W = A_0 \oplus A_i,$$

where  $A_0 \neq A_i$ .

[0078] This results in the operation  $HC_i$  always adding the zeroth auxiliary value  $A_0$  and compensating the first auxiliary value  $A_1$ . As a result, it is possible that all cryptographic operations  $F_i$  are essentially identical, which requires a much smaller processing and/or storage capacity from a processor system with which the method is carried out. In the embodiment of FIG. 8, the operations  $F''_i$  are such adjustments of the original operations  $F_i$ , that these are corrected for the auxiliary value  $A_1$  and in addition combine the tertiary auxiliary value  $W = A_0 \oplus A_1$  with their result. In other words, if  $RD_i \oplus A_1$  is fed to  $F''_i$ , the result will be equal to:

$$FD'_i = F_1(RD_i) \oplus W.$$

[0079] It will be understood by those skilled in the art that the combinatory processes  $AC_i$ ,  $BC_i$  and  $HC_i$  may be carried out in different locations in the cryptographic process P to achieve a comparable or even identical effect.

[0080] FIG. 9 schematically shows a circuit 10 for implementing the method according to the invention. The circuit 10 comprises a first memory 11, a second memory 12 and a processor 13, the memories 11 and 12 and the processor 13 being coupled using a data bus 14. By providing two memories, it is possible each time to carry out a substep of one of the processes P and P\* (see FIG. 4), to store the result of said substep in, e.g., the first memory 11, and from the second memory 12 to transfer a previous interim result from the other process to the processor 13. In this manner, it is possible to

efficiently carry out the alternating computation of substeps of two different processes.

[0081] The payment system schematically shown in FIG. 10 comprises an electronic payment means 1 and a payment station 2. The electronic payment means 1 is, e.g., a so-called smart card, i.e., a card provided with an integrated circuit for storing and processing payment data. The payment station 2 comprises a card reader 21 and a processor circuit 22. The processor circuit 22 may correspond to the circuit 10 of FIG. 9.

[0082] At the beginning of a transaction, the payment means 1 transmits an identification (card identification) ID to the payment station 2. By reference to ~~said the~~ identification, the payment station 2 determines a key which will be used for said transaction. ~~Said The~~ identification ID may be fed as input data X (see the figures 1-3) to a cryptographic process which, on the basis of a master key MK (not shown), produces an identification-dependent transaction key  $K_{ID}$  as output data Y. In accordance with the invention, for this purpose the process shown in the figures FIG. 2 and 3 is used, the master key MK having been converted in advance, using a process R, into a supplementary master key  $MK^*$ . ~~Said The~~ supplementary master key  $MK^*$  is now fed, preferably together with the identification ID, in accordance with FIG. 3, to the supplementary process  $P^*$  in order to reproduce the original master key MK and to derive the transaction key  $K_{ID}$  from the identification ID.

[0083] Although, in the figures FIG. 2 and 3, there is always shown one single supplementary process P\*, there may possibly be used several processes P\*, P\*\*, P\*\*\*, ... in series and/or in parallel to derive the primary key K.

5

[0084] It will be understood by those skilled in the art that many modifications and amendments are possible without departing from the scope of the invention.